Steinmetz, Nadine; Arning, Ann-Katrin; Sattler, Kai-Uwe:

When is Harry Potters birthday? - question answering on linked data

Original published in:	Datenbanksysteme für Business, Technologie und Web / BTW 18. 2019 Rostock. - Bonn : Gesellschaft für Informatik e.V., [2019]. - 289 (2019), p. 565-568.
	ISBN 978-3-88579-683-1
	(GI-Edition. Proceedings / Gesellschaft für Informatik ; 289)
Original published:	2019
ISSN:	1617-5468
DOI:	10.18420/btw2019-46
[*Visited:*	2020-02-27]

T. Grust et al. (Hrsg.): Datenbanksysteme für Business, Technologie und Web (BTW 2019),
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2019 565

When is Harry Potters Birthday? — Question Answering on Linked Data

Nadine Steinmetz[1] Ann-Katrin Arning[1] Kai-Uwe Sattler[1]

Abstract: Question Answering (QA) systems provide a user-friendly way to access any type of knowledge base and especially for Linked Data sources to get insight into semantic information. But understanding natural language, transferring it to a formal query and finding the correct answer is a complex task. The challenge is even harder when the QA system aims to be easily adaptable regarding the underlying information. This goal can be achieved by an approach that is independent from the knowledge base. Thereby, the respective data can be replaced or updated without changes on the system itself. With this paper we present our QA approach and the demonstrator which is able to consume natural language questions of general knowledge (not specific to a topic or domain).

1 Motivation

Question answering (QA) is a research discipline at the intersection of natural language processing (NLP), information retrieval, and database processing aiming at answering questions formulated in natural languages. Though, early QA approaches dating back to the sixties, this field has gotten great attention and research made a significant progress over the last few years. Most well-known examples are personal assistants with voice recognition such as Apple Siri, Amazon Alexa or Microsoft Cortana. These systems are able not only to execute spoken commands but also to answer (simple) questions in natural language. However, answering complex questions beyond asking for facts as well as dealing with ambiguities are still challenging problems.

QA systems work usually in a sequence of the following steps: (1) question parsing and focus detection, (2) question classification, (3) query generation, (4) answer candidate generation (query execution), and (5) result ranking. Furthermore, QA systems make use of structured databases / knowledge bases as sources for answers. Here, particularly RDF databases are useful by allowing a schema-agnostic approach where the schema has not to be known for query generation because all facts are represented by triples. In addition, RDF allows to exploit more advanced semantic concepts such as semantic equivalence, similarity or inference mechanisms. Finally, large collections of Linked Data based on a core set such as DBpedia, Wikidata or YAGO represent a great source for answering a wide range of (not only) factoid questions.

[1] TU Ilmenau, Databases & Information Systems Group, Ilmenau, Germany, *first.last*@tu-ilmenau.de

doi:10.18420/btw2019-46

In our work, we focus on the steps of query generation and answer generation by using a *generic* approach. Compared to existing works our query generation approach is independent from the underlying knowledge base. This is done by representing queries through basic graph patterns whose mapping to knowledge base-specific properties or labels are determined at runtime.

In this paper we demonstrate a prototype implementation of our QA system which is able to answer common (but also complex) natural language questions (in English language) on DBpedia as an example knowledge base.

2 System Architecture

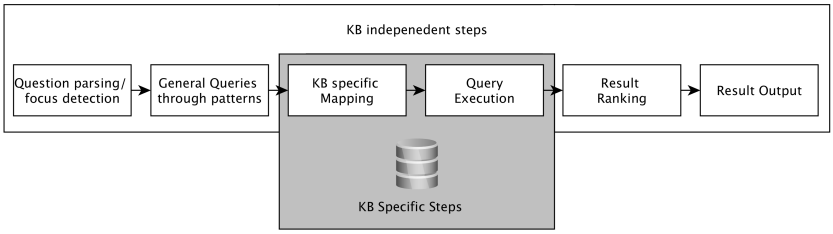

Fig. 1: Overall process of Question Answering

Our approach for question answering consists of seven consecutive steps. Within our algorithm two tasks are dependent on the underlying knowledge base as the concrete properties, entities and ontology classes or categories are requested – as shown in Figure 1. All other steps are independent from the knowledge base.

Question parsing and focus detection. The natural language phrase is analyzed for part-of-speech (POS) and subsequently the question type and the resulting focus are identified.

Generation of general queries with the phrases of the natural language question according to pre-defined patterns. General queries are generated using the extracted phrases from the question and adding the required variables as detected by the previous step.

Mapping subject/predicate/object of the general question. The natural language phrases used for the queries are mapped to the underlying knowledge base to retrieve the actual entities, properties and classes/categories required for the SPARQL query. The general queries are then replaced by executable SPARQL queries using the retrieved resources.

Query execution All generated queries are executed on the underlying knowledge bases and results are retrieved.

Result ranking Each result is scored according to plausibility compared to the identified question type and the expected result type.

Output of the highest ranked SPARQL query and the corresponding result The result of the highest ranked query is provided as assumed correct answer for the question. Our demonstrator also provides all executed queries and the retrieved results together with the computed scores.

Please see [SAS19] for further details on our approach and system architecture.

3 Demonstration

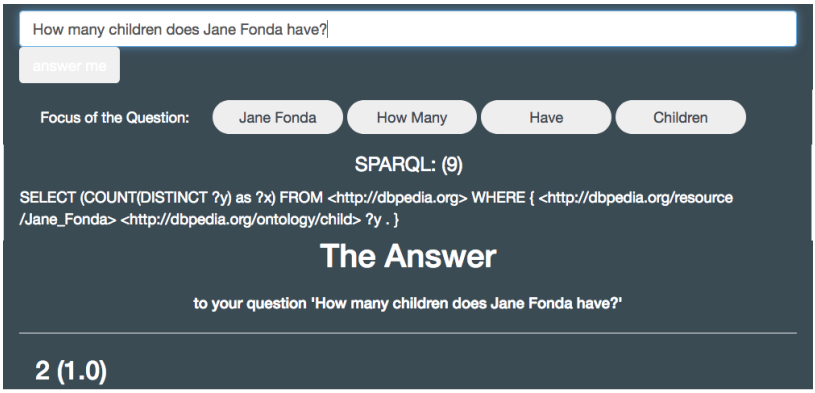

Fig. 2: The system frontend showing the question, the extracted key words and the respective answer.

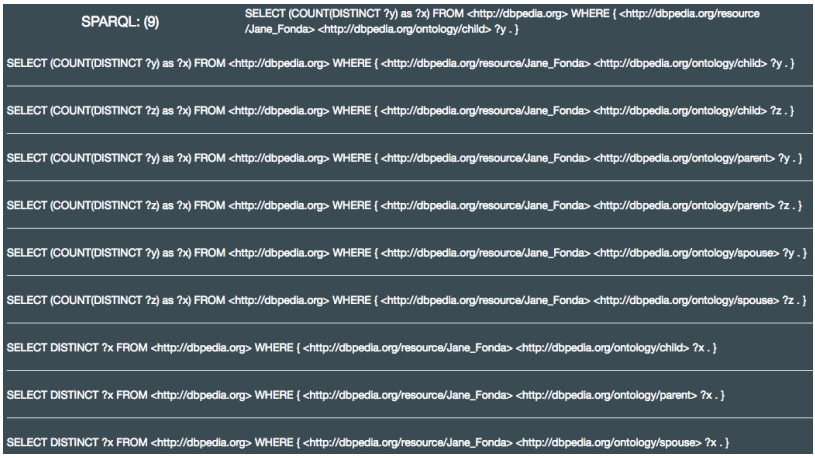

Fig. 3: The system frontend showing all generated SPARQL queries.

For the demonstration of our system the frontend allows a natural language input and provides the result of the most relevant query including more details about the generated queries and the respective answers. Our demonstrator is able to handle different types of questions, such as “How many”, “Who”, “When”, “Where”, “List all”, “Show me”, etc.

The underlying knowledge base is DBpedia. This allows questions on many different topics without restriction on a specific domain.

Figure 2 shows the frontend when asked for “How many children does Jane Fonda have?” and the respective correct answer. The demonstrator also presents the extracted key words resulting from the first step of analyzing the question.

Next, all generated queries for the input question are shown to the user. In this way the user can comprehend the different mapping results for the extracted key words as well as different operators (e.g. COUNT for the given example of questions asking for “How many“). For the given example, each query returns a result from the knowledge base. Figure 4 shows the frontend presenting all retrieved results and the respective score for each result. For the given example the demonstrator shows that the results with the correct answer type (a number for the question “How many”) received the highest overall score.

Fig. 4: The system frontend showing all retrieved results and their scores.

References

[SAS19] Steinmetz, Nadine; Arning, Ann-Katrin; Sattler, Kai-Uwe: From Natural Language Questions to SPARQL Queries: A Pattern-based Approach. In: Datenbanksysteme für Business, Technologie und Web (BTW), 18. Fachtagung des GI-Fachbereichs Datenbanken und Informationssysteme (DBIS), 4.-8.3.2019 in Rostock, Germany. Proceedings. March 2019.